

# 一、填空题（每空 1 分，共 20 分）

1. 十六进制数 80A2H 表示的十进制无符号数为 \_\_\_\_\_，表示的十进制有符号数为 \_\_\_\_\_。
2. 设机器字长为 8 位，则十进制数 -75 的补码的二进制形式是 \_\_\_\_\_。
3. 设寄存器 AX 中的数为 0A95BH，寄存器 BX 中的数为 8CA2H，则指令 ADD AX, BX  
后，AX= \_\_\_\_\_，CF= \_\_\_\_\_。
4. 存储器地址 23E0:2A30H 的段地址、偏移地址和物理地址分别为 \_\_\_\_\_、  
\_\_\_\_\_ 和 \_\_\_\_\_。
5. 在 PC 机存储器中存放的信息如右图所示，试读出 30021H 和  
30022H 单元的字内容，它们分别是 \_\_\_\_\_ 和 \_\_\_\_\_。 30020 H  
30024 H
6. 指令 MOV AX, COUNT[BX][SI] 中，源操作数的寻址方式  
是 \_\_\_\_\_。
7. 假设堆栈段寄存器 SS 的内容为 2250H，堆栈指针 SP 的内容为  
0140H，如果在堆栈中存入 5 个字数据，则 SP= \_\_\_\_\_；如果  
又从堆栈中取出 2 个字数据，则 SP= \_\_\_\_\_。



8. 在 06E4H 单元内有一条两字节指令“jmp short obj”，如指令中转移位移量为 050H，则转向目的地址（有效地址）是\_\_\_\_\_。
9. 在串操作指令中，SI 指向\_\_\_\_\_串，DI 指向\_\_\_\_\_串。
10. 循环控制部分的连续两条指令 DEC CX 和 JNZ NEXT 可以用一条\_\_\_\_\_指令代替。
11. 中断向量表的物理地址范围是\_\_\_\_\_。
12. 模块化设计中定义外部符号的伪操作有：\_\_\_\_\_，\_\_\_\_\_。



## 二、改错题：(共 20 分)

1. 单条指令判断 (注：若指令有错误，指出错误原因即可；每题 1 分)。

- (1) MOV [BX], [SI].
- (2) POP BL .
- (3) ADD BL, AX .
- (4) MOV AL, 2A0H .
- (5) DEC [DI].
- (6) MOV [DX], BL .
- (7) MOV ES, 0A234H .
- (8) CMP BX, AL .
- (9) CMP [BX], 5 .
- (10) ADD BL, [CX+SI] .



2. 判断改错（注：判断下列表述是否正确；若有错误，请在错误位置下画线，并写出正确的表述方式；每题2分）。

(1) 8086/8088 的存储器可分为 4 个段。

(2) 在 80x86 里，操作数的偏移地址和有效地址 EA 是有所区别的。



### 三、简答题（每题 4 分，共 20 分）。

1. 8086 汇编语言提供哪几种数据寻址方式？
2. 试述“`CMP AX, BX`”和“`SUB AX, BX`”两条语句的异同点。
3. 标号有哪三个属性？其含义是什么？
4. 中断响应时由 CPU 自动完成哪些操作？
5. 汇编程序设计中机器指令语句和伪操作语句的区别和作用是什么？



#### 四、分析题：（每题 5 分，共 10 分）。

1. 假设 X 和 X+2 单元的内容为双字数 P，Y 和 Y+2 单元的内容为双字数 Q (X, Y 为低位字)，请说明下面程序段的功能。

mov dx, x+2.

mov ax, x.

add ax, x.

adc dx, x+2.

cmp dx, y+2.

jl l2.

l

(转下页)。



exit ret.

2. 现有程序如下, 请回答: 1) scmp子程序功能是什么? 2) 该程序完成的功能是什么?

stack segment stack 'stack'

dw 100 dup(0)

tos label word

stack ends

data segment

buf1 db 100 dup(?)

buf2 db 200,0,200 dup(0)

msg1 db 'It is good! \$'

msg2 db 'It is bad! \$'

flag db 0

data ends

4)

code segment

assume cs:code, ds:data, ss:stack

start: mov ax, data



```

int 21h.
mov cx, 100
lea si, buf1
lea di, buf2+2
call scmp
cmp flag, 0
jz n
lea dx, msg1
mov ah, 9
int 21h
jmp exit
n: lea dx, msg2
    mov ah, 9
    int 21h
exit: mov ax, 4c00h
      int 21h

```

scmp proc near

```

push ax
cld
cmpsb
jnz r
mov flag, 0ffh
imm 1

```

AH	功 能	调用参数	返 回 参数
09h	显示字符串	DS: DX = 基地址, ‘\$’ 结束字符串	
0Ah	键盘输入到 缓冲区	DS: DX = 缓冲区首 地址, (DS: DX) = 缓冲区最大字符数	(DS: DX+1) = 实 际输入的字符数



## 五. 设计题：(每题 10 分，共 30 分)

1. 请编写完整的可执行程序，计算  $Z=(Y+5) \times X + 20$ ，X、Y 为无符号字节数据，Z 为

符号字数据。

2. 编写一个统计指定缓冲区中正数、负数和 0 的个数的子程序。要求在主程序中调用

子程序，将缓冲区 buf1、buf2、buf3 中的数分别进行统计。

3. 在首地址为 ARY 的子数组中，存放了 100H 个 16 位补码数，试编写程序，求出它

的平均值将整数部分放在 V 字单元中，并求出数组中有多少个数大于此平均值(整数部分)

将结果放在 COUNT 字单元中。



## 一、填空题（20分，每空1分）

- 1、已知字节操作数  $X = -23H$ 、 $Y = 56H$ ，则  $[X]_B - [Y]_B = [ \dots ]_B$ 。
- 2、已知指令执行前  $AX$  内容为  $3412H$ ，执行  $CMP AX, 5432H$  指令后  $ZF = \underline{\hspace{2cm}}$ ,  $SF = \underline{\hspace{2cm}}$ 。
- 3、对 IO 设备的控制或数据传输实质上是对        的数据读写。
- 4、条件转移指令只能使用        寻址方式。
- 5、8086CPU 执行  $POP$  指令时，堆栈指针  $SP$  会自动修改，修改后  $SP = \underline{\hspace{2cm}}$ 。
- 6、X86 实模式下，段大小不超过  $64K$ ，这是由于        长度为 16 位。
- 7、段间  $CALL$  与转移指令的不同之处是 CPU 会自动将        保存在堆栈中。
- 8、汇编程序设计时参数传送一般通过：        和        两种途径。
- 9、 $ORG$  伪操作用来设置当前        的值，即用于分配后续数据、指令的存储器单元的       。
- 10、跳转表中转移地址的数据定义时，若用  $DW$  定义，则对应       ，若用  $DD$  定义，则对应       。
- 11、宏调用是使用用户自定义的宏指令，汇编展开时将宏调用用        来代替。
- 12、中断过程包括：       、       以及中断返回三个步骤。
- 13、用 debug 调试程序时，寄存器和存储单元内容如下所示，则：  
① 执行  $pop ax$  指令后， $ax = \underline{\hspace{2cm}}$ ；  
② 执行  $mov ax, [bx+2]$  指令后， $ax = \underline{\hspace{2cm}}$ ；  
③  $077EEH$  存储单元的字数据是       。

```
-r  
AX=077E  BX=0000  CX=01F6  DX=0000  SP=0004  BP=0000  SI=0000  DI=0000  
DS=077E  ES=076E  SS=077E  CS=0782  IP=000A  NV UP EI PL NZ NA PO NC  
-d 077E:0000  
077E:0000  82 07 AD 01 31 24 73 6F-6E 67 32 24 73 6F 6E 67
```



## 二、判断改错题 (20 分)

1、单条指令判断。如果指令错，写出错误原因。(每题 1 分)

- (1) MOV DS, SEG ARRAY
- (2) PUSH AL
- (3) CMP [SI], BYTE PTR [BX]
- (4) DIV BX, DX
- (5) SAL DX, 3
- (6) IN AL, 100H
- (7) LEA DI, OFFSET ARRY
- (8) MOVS ES:[SI], DS:[ DI]
- (9) JNZ DWORD PTR [BX]
- (10) INC [BX]

2、改错(写出正确的表述方式；每题 2 分)

- (1) 80X86 的机器指令格式一般为：段寄存器名：[基址+变址\*比例因子+位移量]。
- (2) 无条件转移 (JMP) 指令只可以用于段内转移。
- (3) 带立即数的子程序返回指令中，立即数主要是为了向主程序传输返回参数值。
- (4) 标号在转移或 CALL 指令的操作数字段出现时，表示变量的存储单元地址。
- (5) BIOS 和 DOS 功能调用时，使用开中断指令 STI，采用中断方式实现功能调用。

## 三、简答题 (20 分，每小题 4 分)

- 1、请说明 MOV 和 MOVS 指令的相同和不同是什么？
- 2、请具体说明汇编程序返回 DOS 操作系统的两种实现方法。
- 3、请简述宏调用与子程序调用的区别和各自的优缺点。
- 4、中断响应时由 CPU 自动完成哪些操作？
- 5、请以 MOV AX, DS:[SI] 的源操作数为例画图说明实模式下数据读取过程。

## 四、分析题 (10 分，每小题 5 分)

- 1、假设 DS=076AH，请画出下面数据段中变量的存储单元分配及其初始化值。用十六进制数表示存储单元内容的初始化值，标出存储单元地址和变量位置。



# 西安交通大学考试题

```
data segment
    org 0200h
    m1  db 86h, 80h
    m2  db 2 dup('86')
    m3  dw m1
    len dw $-m1
data ends
```

2、读下面数组累加程序，并回答问题。

- (1) 填表, 当执行到子程序 proadd 中的指令 “`next: add ax, [di]`” 时堆栈的使用情况, 堆栈单元内容可用符号名写出。

(2) 说明程序中下划线指令(短虚线框中)的源操作数寻址方式。

stack segment at 0A000h ; 堆栈段段地址从A000h开始

堆栈单元内容	堆栈单元 段地址:偏移地址
	A000:0000
	...
	...
	A000:00F2
	A000:00F4
	A000:00F6
	A000:00F8
	A000:00FA
	A000:00FC
	A000:00FE
tos	A000:0100

mov ax, data : 寻址方式

push ax ; 寻址方式

lea di, ary : 寻址方式



```

code1 segment
    assume cs:code1
proadd proc far
    push ax
    push cx
    push di
    lea di, ary
    mov cx, count
    xor ax, ax
next: add ax, [di]
        add di, 2
        loop next
        mov sum, ax
        pop si
        pop cx
        pop ax
        ret
proadd endp
code1 ends
end start

```

## 五、编程题 (30 分, 每小题 10 分, 要求编写完整的 8086 程序)

1、请写出计算  $Z = (X^2 + 2XY + Y^2) / 32$  的可执行程序, 假设运算中结果没有溢出。其中, X、Y、Z 为有符号字数据。

2、在内存缓冲区中有一个字符串 ARRY, 缓冲区第一个单元中存储该字符串的长度, 请编写程序满足以下要求:

- (1) 采用子程序调用的程序结构, 并使用堆栈传递变量地址;
- (2) 在子程序中查找字符串的第一个“空格(20H)”字符并删除, 删除成功后将后续字符向前移动, 同时修改字符串长度。

3、请编写满足以下功能的程序。程序运行后:

- (1) 清屏(黑底白字属性=07H);
- (2) 在屏幕上开一窗口(红底黄字属性=4EH), 左上角坐标(06H, 14H), 右下角坐标(14H, 2EH);
- (3) 在窗口内显示如右图所示菜单;
- (4) 当按下键盘键“4”时: 退出程序, 返回到 DOS 环境;
- (5) 当按下其它按键时: 继续等待。

MAIN MENU

- (1) Edit
- (2) Save
- (3) Print
- (4) Quit

附录: 部分 BIOS/DOS 系统功能调用参数

INT n	AH	功 能	调用参数	返 回 参数
10H	06H	初始化屏幕窗口或上卷	AL=上卷行数(AL=0,清窗口) BH=卷入行属性 CH=窗口左上角行号(Y坐标) CL=窗口左上角列号(X坐标) DH=窗口右下角行号(Y坐标) DL=窗口右下角列号(X坐标)	
10H	02H	设置光标位置	BH=页号 DH=行号(Y坐标), DL=列号(X坐标)	
10H	0EH	显示字符 (光标迁移)	BH=显示页 AL=字符, BL=前景色(黄色=13H)	
21H	07H	从键盘输入 1 个字符, 不回显		AL=字符码
21H	08H	读键盘状态		AL=0FFH 有键入 AL=00 无键入

